



## CpSc 360: Distributed and Network Programming

# Network Basics

James Wang



\* Some materials in this lecture note, such as figures and diagrams, were drawn from Internet



## Network

Hardware and software data communication systems.

Network Layers:

The OSI seven layer model attempts to provide a way of partitioning any computer network into independent modules from the lowest (physical) layer to the highest (application) layer. Many different specifications exist at each of these layers.

Network Classification:

- local area network (LAN)
- metropolitan area network (MAN)
- wide area network (WAN)



## LAN - Local Area Network

connects computers that are physically close together (< 1 mile).

- high speed
- multi-access through shared wires.

Technologies:

- Ethernet 10 Mbps, 100Mbps, 1000 Mbps
- Token Ring 16 Mbps
- FDDI 100 Mbps

Each department maintains one or more LANs.



## WAN - Wide Area Network

connects computers that are physically far apart. "long-haul network".

- typically slower than a LAN.
- typically less reliable than a LAN.
- point-to-point communication through many routers/switches.

Technologies:

- telephone lines.
- Satellite communications.
- Fiber network.



## MAN - Metropolitan Area Network

Larger than a LAN and smaller than a WAN

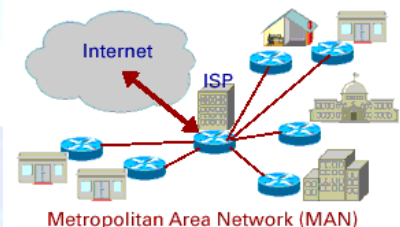
- Can be viewed as a small region WAN.
- multi-access network.
- Many routers and switches are needed.

Technologies:

- Coaxial cable to connect LANs in different buildings.
- Microwave to connect LANs through air.



## MAN Example





## OSI 7 Layer Model:

- |   |                     |                      |
|---|---------------------|----------------------|
| 7 | <b>Application</b>  | High level protocols |
| 6 | <b>Presentation</b> | ↑<br>-----<br>↑      |
| 5 | <b>Session</b>      |                      |
| 4 | <b>Transport</b>    |                      |
| 3 | <b>Network</b>      |                      |
| 2 | <b>Data-Link</b>    |                      |
| 1 | <b>Physical</b>     | Low level protocols  |



## The Physical Layer

- Responsibility:**
  - transmission of raw bits over a communication channel.
- Issues:**
  - mechanical and electrical interfaces
  - time per bit
  - distances



## The Data Link Layer

### Data Link Control:

- Responsibility:**
  - provide an error-free communication link
- Issues:**
  - framing* (dividing data into chunks)
    - header & trailer bits
  - addressing

10110110101 → 01100010011 → 10110000001 →

### The MAC sublayer:

- Medium Access Control - needed by multiaccess networks.**
- MAC provides DLC with “virtual wires” on multiaccess networks.



## The Network Layer

- Responsibilities:**
  - path selection between end-systems (routing).
  - subnet flow control.
  - fragmentation & reassembly
  - translation between different network types.
- Issues:**
  - packet headers
  - virtual circuits



## The Transport Layer

- Responsibilities:**
  - provides virtual end-to-end links between peer processes.
  - end-to-end flow control
- Issues:**
  - headers
  - error detection
  - reliable communication



## The Session Layer

- Responsibilities:**
  - establishes, manages, and terminates sessions between applications.
  - service location lookup
- Many protocol suites do not include a session layer.**
  - For instance, TCP/IP protocol.





## The Presentation Layer

- Responsibilities:
  - data encryption
  - data compression
  - data conversion
- Many protocol suites do not include a Presentation Layer.



## The Application Layer

- Responsibilities:
  - anything not provided by any of the other layers
- Issues:
  - application level protocols
  - appropriate selection of "type of service"

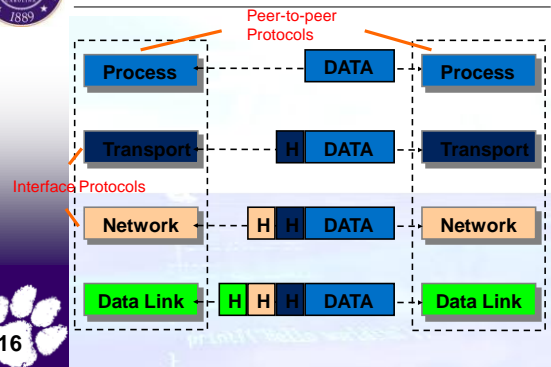


## Layering & Headers

- Each layer needs to add some control information to the data in order to do its job.
- This information is typically prefixed to the data before being given to the lower layer.
- Once the lower layers deliver the data and control information - the peer layer uses the control information.



## Headers



## What are the headers?

**Physical:** no header - just a bunch of bits.

**Data Link:**

- address of the receiving endpoints
- address of the sending endpoint
- length of the data
- checksum.



## Network layer header - examples

- protocol suite version
- type of service
- length of the data
- packet identifier
- fragment number
- time to live
- protocol
- header checksum
- source network address
- destination network address



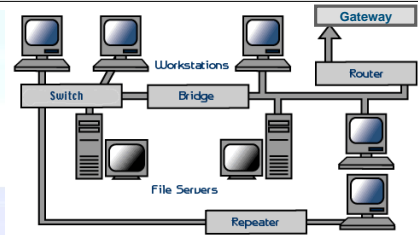


## Network Summary

- ✿ **Data-Link:** communication between machines on the same network.
- ✿ **Network:** communication between machines on possibly different networks.
- ✿ **Transport:** communication between processes (running on machines on possibly different networks).



## Network Connectors



- ✿ **Repeater:** physical layer
- ✿ **Bridge:** data link layer
- ✿ **Router:** network layer
- ✿ **Gateway:** network layer and above.



## Repeater

- ✿ Copies bits from one network to another
- ✿ Does not look at any bits
- ✿ Allows the extension of a network beyond physical length limitations



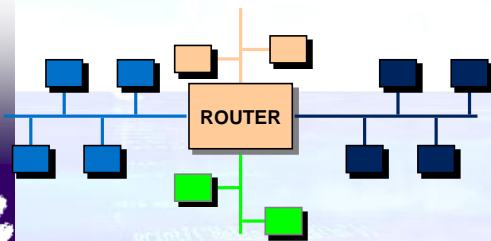
## Bridge

- ✿ Copies frames from one network to another
- ✿ Can operate selectively - does not copy all frames (must look at data-link headers).
- ✿ Extends the network beyond physical length limitations.



## Router

- ✿ Copies packets from one network to another.
- ✿ Makes decisions about what *route* a packet should take (looks at network headers).



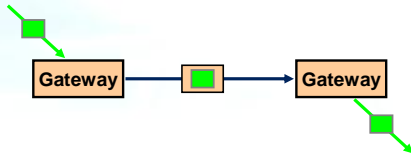
## Gateway

- ✿ Operates as a router
- ✿ Data conversions above the network layer.
- ✿ **Conversions:**
  - encapsulation - use an intermediate network
  - translation - connect different application protocols
  - encryption - could be done by a gateway





## Encapsulation and Translation



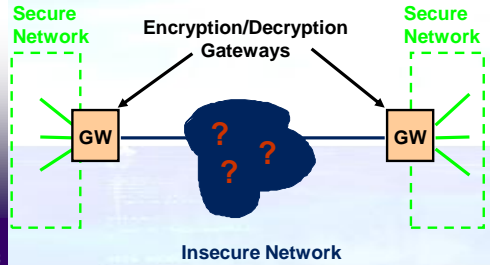
- Provides service connectivity even though intermediate network does not support protocols.



- Translate from green protocol to brown protocol



## Encryption gateway



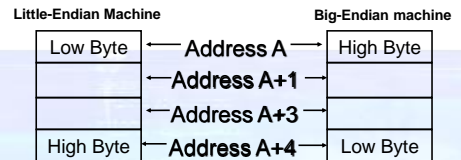
## Hardware vs. Software

- Repeaters are typically hardware devices.
- Bridges can be implemented in hardware or software.
- Routers & Gateways are typically implemented in software so that they can be extended to handle new protocols.
- Many workstations can operate as routers or gateways.



## Byte Ordering

- Different computer architectures use different byte ordering to represent multibyte values.
- 32 bit integer:



## Machines with Different Endian

Machine	Endian
===== Compaq	===== Little
Cray	Big
IBM	Big
MAC	Big
PC	Little
SGI	Big
Sun	Big
Vax	Little



## Byte Order and Networking

- Suppose a Big Endian machine sends a 16 bit integer with the value 2:

0000000000000010

- A Little Endian machine will think it got the number 512:

0000010000000000





## Network Byte Order

- Conversion of application-level data is left up to the presentation layer.

**But hold on !!!**

- How do lower level layers communicate if they all represent values differently? (data length fields in headers)

- A fixed byte order is used (called **network byte order**) for all control data.



## Modes of Service

- Connection-oriented vs. connectionless**

- A connection-oriented service includes the establishment of a logical connection between 2 processes.

- establish logical connection

- transfer data

- terminate connection.

- Connectionless services involve sending of independent messages.

- Sequencing**

- Sequencing provides support for an order to communications.

- A service that includes sequencing requires that messages (or bytes) are received in the same order they are sent.



## Modes of Service (continue)

- Error-control**

- Some services require error detection (it is important to know when a transmission error has occurred).

- Checksums provide a simple error detection mechanism.

- Error control sometimes involves notification and retransmission.

- flow-control**

- Flow control prevents the sending process from overwhelming the receiving process.

- Flow control can be handled a variety of ways - this is one of the major research issues in the development of the next generation of networks (ATM).



## Modes of Service (continue)

- byte stream vs. message based**

- Byte stream implies an ordered sequence of bytes with no message boundaries.

- Message oriented services provide communication service to chunks of data called datagrams.

- full-duplex vs. half-duplex.**

- Full-Duplex services support the transfer of data in both directions.



- Half-Duplex services support the transfer of data in a single direction.



## End-to-End vs. Hop-to-Hop

- Many service modes/features such as flow control and error control can be done either:

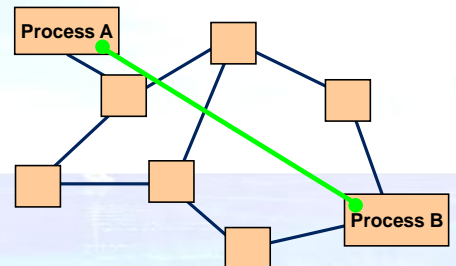
- End-to-End: between endpoints of the communication

OR

- Hop-to-Hop: between every 2 nodes on the path between the endpoints

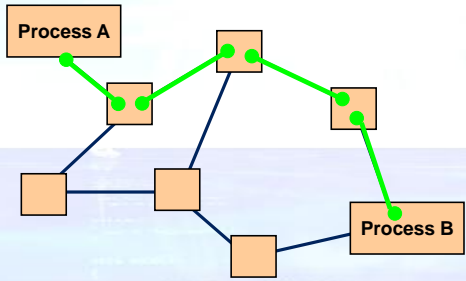


## End-to-End





## Hop-by-Hop



## Buffering

- Buffering can provide more efficient communications.
- Buffering is most useful for byte stream services.



## Addresses

- Each communication endpoint must have an address.
- Consider 2 processes communicating over an internet:
  - the network must be specified
  - the host (end-system) must be specified
  - the process must be specified.



## Addresses at Layers

- Physical Layer: no address necessary
- Data Link Layer - address must be able to select any host on the network.
- Network Layer - address must be able to provide information to enable routing.
- Transport Layer - address must identify the destination process.

